



人事管理系统

Personnel Management System(PMS)

软件（结构）设计说明[SDD]文档

Version: 1.0.1

2024-5-20

All Rights Reserved

教师：余仲星

队员：马豪 李晴 高潼 全鑫 耿尊至

目录

1 引言	3
1.1 标识	3
1.2 系统概述.....	4
1.3 文档概述.....	4
1.4 基线	4
2 引用文件.....	5
3 CSCI 级设计决策.....	5
3.1 关于员工信息管理的设计决策:	5
3.2 关于员工状态管理的设计决策:	5
3.3 为满足安全性、保密性、私密性需求而选择的方法:	6
4 CSCI 体系结构设计.....	7
4.1 体系结构.....	7
4.1.1 程序(模块)划分	7
4.1.2 程序(模块)层次结构关系	8
4.2 全局数据结构说明.....	9
4.2.1 常量.....	9
4.2.2 变量.....	9
4.2.3 数据结构.....	10
4.3 CSCI 部件.....	11
4.4 执行概念.....	14
4.4 执行概念.....	14
4.5 接口设计.....	16
4.5.1 接口标识与接口图.....	16
5 CSCI 详细设计.....	18
5.1 员工信息管理模块.....	18
5.1.1 概述.....	18
5.1.2 软件配置项设计中的约束、限制或非常规特征	18
5.1.3 编程语言和使用理由.....	19
5.1.4 过程式命令列表.....	19
5.1.5 输入输出和数据元素说明.....	19
5.1.6 逻辑说明.....	20
5.2 员工工资管理模块.....	20
5.2.1 概述.....	20
5.2.2 软件配置项设计中的约束、限制或非常规特征	20
5.2.3 编程语言和使用理由.....	21
5.2.4 过程式命令列表.....	21
5.2.5 逻辑说明.....	21
5.3 查询与统计.....	22
5.3.1 概述.....	22
5.3.2 软件配置项设计中的约束、限制或非常规特征	22
5.3.3 编程语言和使用理由.....	22
5.3.4 过程式命令列表.....	22

5.3.5 逻辑说明.....	22
6 需求的可追踪性.....	23
6.1CSCI 需求可追踪性.....	23
6.1.1 查询模块.....	23
6.1.2 员工信息修改模块.....	23
6.1.3 员工管理模块.....	24
6.1.4 提交申请模块.....	24
6.2 CSCI 需求到它被分配给的软件配置项的可追踪性.....	24
6.2.1 查询功能.....	24
6.2.2 员工信息修改功能.....	24
6.2.3 员工管理功能.....	24
6.2.4 提交申请功能.....	24
7 注解	25
7.1 背景信息.....	25
7.2 术语和定义.....	25
7.3 缩略语列表.....	25
附录	25

1 引言

1.1 标识

本文档适用于基于 web 的人事管理系统的设计，该系统的完整标识如下：

- 标识号：PMS 1.0.1
- 标题：人事管理系统
- 缩略词语：PMS 1.0.1(Personnel Management System)
- 版本号：1.0.1
- 发行号：20240506

1.2 系统概述

该项目是一个基于 web 的人事管理系统，具备更丰富的功能，提供更友好、更直观的用户界面，统能够自动化、集中化地管理员工信息，实现快速、准确的数据录入、修改和查询，提供全面的、实时的数据分析，以及对员工信息的深度挖掘和预测功能，并且保持系统的稳定性和可靠性。

该系统支持用户修改个人信息，查询员工普通信息，查询工资，绩效情况，管理员用户管理员工状态。此外，该系统还允许用户提交各种申请，上传相关文件。

该系统的当前运行现场主要是在互联网上，用户可以通过访网页访问使用。未来，作为开发方，我们计划将该系统扩展为应用程序，在移动端(Android/Mac)上，以满足用户的多样化需求。

本文档适用于人事管理系统的设计和开发，其他有关文档包括需求规格说明书、用户手册、测试计划和测试报告等。

1.3 文档概述

本文档是基于 web 的人事管理系统的软件(结构)设计说明(SAD)文档。本文档的主要目的是为开发人员提供关于系统的结构设计和实现细节的详细信息，以便他们能够更好地理解系统的内部结构和交互方式，并在系统开发的不同阶段中进行有效的协调和沟通。

本文档包括系统的总体设计，包括系统架构、模块划分和接口设计等方面的内容。此外，本文档还提供了关于每个模块的详细设计信息，包括功能需求、接口、数据结构和算法等方面的内容。

本文档的使用需要遵守相应的保密性或私密性要求。未经授权，禁止任何人员泄露、使用或复制本文档的任何部分。

1.4 基线

本系统设计说明书所依据的设计基线是需求规格说明书(SRS)，其中包括了系统的功能需求、性能需求、可靠性需求、用户界面需求、安全需求、可维护性需求等方面的详细描述。此外，还有概要设计和详细设计文档，它们描述了软件

系统的体系结构、模块划分、接口定义、数据结构、算法、数据库设计等方面的信息，为本系统设计说明书的编写提供了基础和指导。同时，本设计说明书还遵循了相关的行业标准和最佳实践，包括但不限于软件工程领域的 IEEE 标准、ISO 标准等。在编写过程中，还参考了类似的应用程序的设计和实现方案，从中汲取经验和教训，以确保本系统设计方案的有效性和可行性。

2 引用文件

本文档引用以下文件：

1. 人事管理系统可行性分析（研究）报告（FAR）
版本号：1.0.1，日期：2024 年 3 月 25 日
2. 人事管理系软件需求规格（SRS）文档
版本号：1.0.1，日期：2023 年 3 月 25 日

除上述文件外，本文档未引用任何其他文档。所有引用文档的最新版本可通过正常的供货渠道获得。

3 CSCI 级设计决策

3.1 关于员工信息管理的设计决策：

输入输出：

输入：员工的基本信息、学历信息、婚姻状况、其他相关信息。

输出：录入或修改后的员工信息。

设计信息录入接口，允许用户添加或修改员工信息：

与用户接口：提供表单界面，用户可以输入或选择相应的信息。

与数据库接口：设计合适的数据库表结构和存储过程，以便存储和检索员工信息。

设计信息修改接口，允许用户实时更新员工信息：

与用户接口：提供信息修改功能，用户可以随时更新员工信息。

与数据库接口：实现数据库的更新操作，确保修改后的信息立即生效。

设计修改记录追溯功能，记录每次信息修改的详细信息：

与用户接口：提供修改记录的查看功能，用户可以追溯信息的变更历史。

与数据库接口：设计修改记录的数据库表结构，用于存储每次修改的时间、操作人员和修改内容。

3.2 关于员工状态管理的设计决策：

输入输出：

输入：员工的状态变更（如转出、辞职、退休等）。

输出：更新后的员工状态信息。

设计状态标记功能，允许管理员对员工状态进行标记：

与用户接口：提供状态标记的操作界面，管理员可以轻松地进行状态标记。

与数据库接口：实现状态的更新操作，将员工状态信息存储到数据库中。

设计状态展示功能，以明显的方式展示员工的当前状态：

与用户接口：在员工信息页面突出显示员工状态，方便管理员快速了解员工的工作状态。

与数据库接口：从数据库中检索员工状态信息，并将其展示在用户界面上。

其他功能的设计决策可以类似地进行，包括工资管理、查询与统计以及系统安全性等方面的设计决策。

3.3 为满足安全性、保密性、私密性需求而选择的方法：

用户身份验证：

- 采用强密码策略和多因素认证机制，确保只有授权用户能够访问系统。
- 实现密码加密存储，使用哈希算法，以保护用户密码不被泄露。

访问控制和权限管理：

- 根据用户角色和职责定义不同的访问权限，限制用户对敏感数据和功能的访问。
- 实施最小权限原则，确保用户仅能访问其工作所需的数据和功能。

数据加密和保护：

- 对敏感数据进行加密存储和传输，使用加密算法（如 AES）。
- 实施数据掩码和脱敏技术，以保护在显示和报表中的敏感信息。

安全审计和监控：

- 记录和监控系统的所有操作和访问尝试，包括成功和失败的登录尝试。
- 定期审查安全日志，以便及时发现和响应潜在的安全威胁。

数据备份和灾难恢复：

- 定期备份关键数据，确保在数据丢失或损坏时能够迅速恢复。
- 制定灾难恢复计划，以应对系统故障或数据泄露等紧急情况。

网络安全措施：

- 实施防火墙和入侵检测系统（IDS），以保护系统免受外部攻击。
- 使用安全套接字层（SSL）或传输层安全性（TLS）协议，加密数据传输。

用户教育和意识培训：

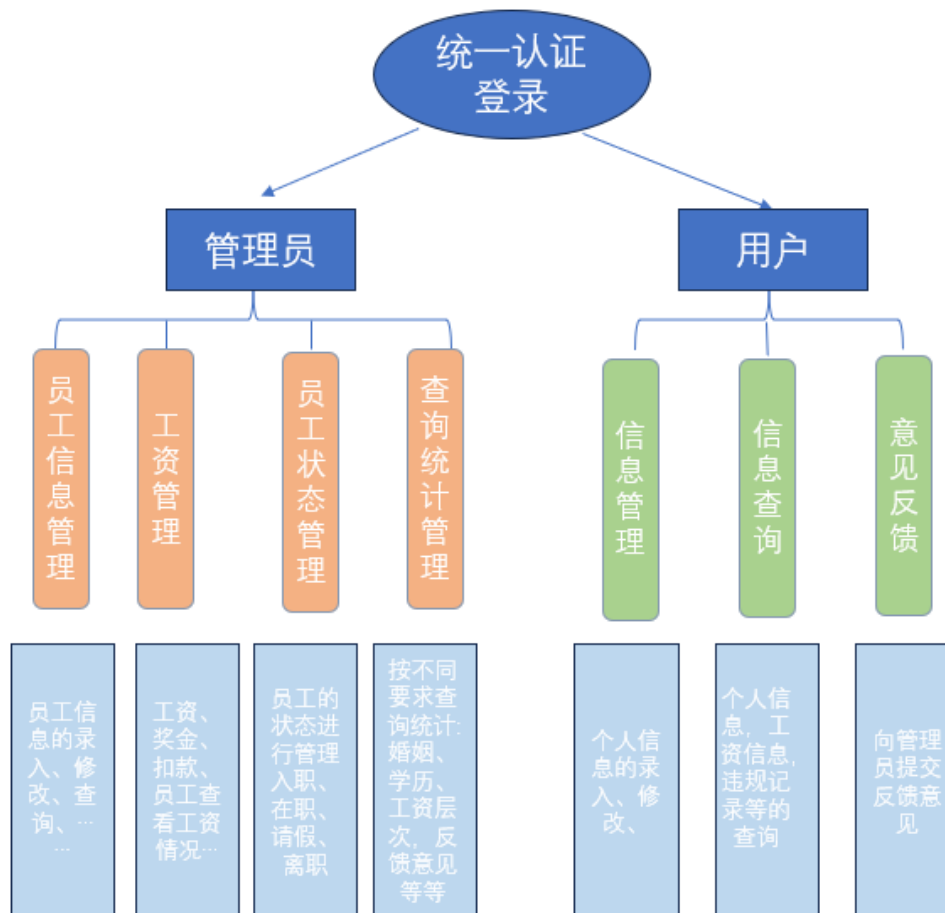
- 提供安全意识培训，教育用户识别和防范潜在的安全威胁，如钓鱼攻击。
- 强调用户对个人信息保护的责任，鼓励使用强密码和个人安全最佳实践。

4 CSCI 体系结构设计

4.1 体系结构

4.1.1 程序(模块)划分

为了详细的说明我们的 PMS 程序结构，我们采用下述图来说明，具体解释如下：



图表 1

统一认证:

这是系统的入口点,确保只有经过验证的用户才能访问系统。包括多因素认证来增加安全性。

管理控制:

权限管理: 确定哪些用户可以访问系统的哪些部分。包括对文件、数据和其他资源的访问控制。

角色管理: 分配用户角色,如 HR 管理员、部门经理或普通员工,每个角色都有不同的权限。

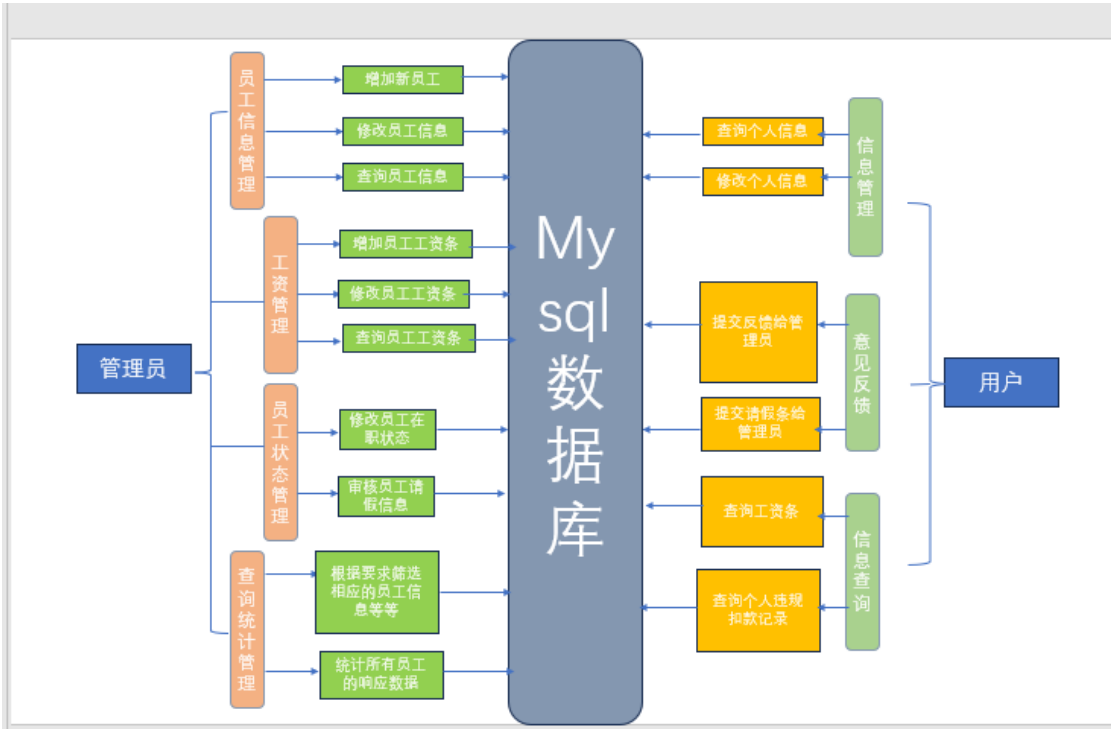
安全策略: 设定系统的安全规则,如密码策略、数据加密和防病毒措施。

用户:

内部人员: 公司的正式员工, 需要使用系统来记录时间、查看工资单或更新个人信息。
合作伙伴: 与公司合作的外部机构或个人, 需要访问特定的项目或报告。
客户: 可能指的是使用公司服务的外部客户, 需要访问账户信息或服务历史。

4.1.2 程序(模块)层次结构关系

用一系列图表列出本 CSCI 内的每个程序(包括每个模块和子程序)之间的层次结构与调用关系。



图表 2

MySQL 数据库:作为人事管理系统的核心, 这个数据库负责存储所有与人力资源相关的数据。它不仅包含员工的基本信息, 包括合同细节、工作历史、培训记录、绩效评估结果等。数据库的设计确保了数据的一致性、完整性和安全性。

数据表:学生信息 (员工信息):在 PMS 中, 包含员工的编号、姓名、性别、出生日期、联系方式、住址、教育背景、工作经验等。它跟踪员工的职位变动、薪资变化和其他重要的职业里程碑。

课程信息 (培训课程信息):记录公司提供的所有培训课程的详细信息, 包括课程编号、名称、描述、教材、培训日期、培训地点和任课教师。它帮助 HR 部门管理员工的职业发展和技能提升。

教师信息 (培训师信息):在企业环境中, 包括内部培训师或外部顾问的详细信息, 如专业领域、资格证书、联系方式和以往的培训效果评价。这有助于选择合适的培训师来满足特定培训需求。

选课信息 (员工培训记录):这个表记录员工报名参加的培训课程、课程进度、完成情况和评

估结果。这有助于监控员工的学习进度和评估培训的有效性。

用户:这个部分代表了所有使用 PMS 的用户,包括:HR 管理员:负责系统的日常运营,包括更新员工记录、生成报告和监控系统安全。

员工:可以访问自己的个人信息、申请假期、查看培训课程和提交绩效自评。

合作伙伴:包括外部培训机构或顾问,他们需要访问系统来管理与公司的合作项目。

客户:这指的是需要公司服务的外部客户,他们可以通过系统访问服务记录或提交查询。

4.2 全局数据结构说明

4.2.1 常量

1. **NAME_LENGTH:** 员工名字的长度限制,数据类型为整型,通常设定为 2 到 15 之间。
2. **AGE_RANGE:** 员工年龄范围,数据类型为整型,通常设定为 18 到 65 之间。
3. **PHONE_LENGTH:** 电话号码范围,数据类型为整型,通常设定为 11。
4. **WORK_AGE_RANGE:** 员工工龄范围,数据类型为整型,通常设定为 0 到 50。
5. **USER_SIZE:** 公司员工上限,数据类型为整形,通常设定为 200 到 500。
6. **MAX_SALARY:** 员工薪资的最大值,数据类型为浮点型,通常设定为一个合理的最大值,如 1000000 (单位:元)。
7. **MIN_SALARY:** 员工薪资的最小值,数据类型为浮点型,通常设定为一个合理的最小值,如 2000 (单位:元)。
8. **MAX_WORK_HOURS:** 员工一周工作小时数的最大值,数据类型为整型,通常设定为 40 小时。
9. **MIN_WORK_HOURS:** 员工一周工作小时数的最小值,数据类型为整型,通常设定为 20 小时。
10. **MAX_VACATION_DAYS:** 员工年假的最大天数,数据类型为整型,通常设定为 15 天。
11. **MIN_VACATION_DAYS:** 员工年假的最小天数,数据类型为整型,通常设定为 5 天。
12. **DEFAULT_DEPARTMENT:** 默认部门名称,数据类型为字符串,通常设定为“未分配”或“其他”等。
13. **DEFAULT_POSITION:** 默认职位名称,数据类型为字符串,通常设定为“员工”或“普通员工”等。
14. **TAX_RATE:** 税率,数据类型为浮点型,用于计算员工的税前工资,通常设定为一个合理的税率,如 0.1 (表示 10%的税率)。
15. **INSURANCE_RATE:** 社保费率,数据类型为浮点型,用于计算员工的社保费用,通常设定为一个合理的费率,如 0.08 (表示 8%的社保费率)。

4.2.2 变量

1. **users:** 当前用户的信息,例如员工号、员工姓名、电话、电子邮件地址、性别等。
2. **ads:** 当前管理员信息,例如管理员号、管理员姓名、电话等。

3. **payRolls:** 当前员工的工资信息，例如工资单号、员工号、绩效、基本工资等。
4. **informations:** 当前员工/管理员信息，包括姓名、性别、电话、账号、密码等。
5. **cur:** 当前所处的页面号，用于显示高亮页面选项。
6. **db:** 数据库信息，用于连接数据库。
7. **sql:** 内嵌 sql 语句变量，用于对数据库执行 sql 操作。
8. **option:** 前端选项变量，用于从前端获取记录对应的选项。
9. **id:** 用户号，用于在数据库中进行检索。
10. **result:** 用于存储数据库查询的中间结果。
11. **data:** 用于存储处理后待返回的数据库查询结果。
12. **attribute:** 用于获取从前端获取的相应查询属性，比如按婚姻状况查询或按性别查询。
13. **error:** 存储相应的错误信息，如用户名不存在，密码错误等。

4.2.3 数据结构

4.2.3.1 用户类 User

保存用户必要信息，完成登录检索等功能

```
class User(models.Model):
    user_id = models.CharField(max_length=20, primary_key=True) # 工号
    user_password = models.CharField(max_length=20) # 密码
    user_gender = models.BooleanField() # 性别
    user_name = models.CharField(max_length=6, default='') # 姓名
    user_academy = models.IntegerField(default=1) # 学历
    user_age = models.IntegerField(default=1) # 年龄
    user_workage = models.IntegerField(default=1) # 工龄
    user_nation = models.CharField(max_length=20, default='') # 民族
    user_times = models.IntegerField(default=1) # 违约次数
    user_phone = models.CharField(max_length=11, default='') # 电话
    user_email = models.CharField(max_length=25, default='') # 邮箱
    user_marriage = models.CharField(max_length=25, default='') # 婚姻状况
    user_post = models.CharField(max_length=25, default='') # 岗位
    user_statue = models.CharField(max_length=25, default='') # 状态
```

4.2.3.2 管理员类 Administrator

保存管理员必要信息，完成登录等功能

```
class Administrator(models.Model):
    ad_id = models.CharField(max_length=20, primary_key=True)
    ad_name = models.CharField(max_length=6)
    ad_password = models.CharField(max_length=20)
    ad_gender = models.BooleanField()
    ad_phone = models.CharField(max_length=11, default='')
    ad_email = models.CharField(max_length=25, default='')
```

4.2.3.3 违约类 default

保存违约信息，完成相应的工资扣款

```
class Default(models.Model):
    default_id = models.IntegerField(primary_key=True)
    default_reason = models.CharField(max_length=200)
    default_id_student = models.CharField(max_length=20)
    default_id_administrator = models.CharField(max_length=20)
    default_time = models.DateTimeField(default='2000-11-11 19:00:00')
```

4.2.3.4 建议类 Complaint

保存建议信息，用于向管理员展示建议信息

```
class Complaint(models.Model):
    complaint_id = models.IntegerField(primary_key=True)
    complaint_time = models.DateTimeField(default='2000-11-11 19:00:00')
    complaint_content = models.CharField(max_length=300)
    complaint_user_id = models.CharField(max_length=20)
```

4.2.3.5 工资类 Payroll

保存工资信息，用于添加、修改、删除员工工资

```
class Payroll(models.Model):
    payroll_id = models.IntegerField(primary_key=True)
    user = models.ForeignKey('User', to_field='user_id', on_delete=models.CASCADE)
    month = models.IntegerField()
    basicSalary = models.FloatField()
    bonus = models.FloatField() # 绩效
    deductions = models.FloatField() # 扣款信息
```

4.3 CSCI 部件

4.3.1 软件配置项及用途：

1. 员工信息管理模块：用于管理公司员工的基本信息，包括姓名、性别、年龄、部门、职位、联系方式等。
2. 工资管理模块：用于管理员工的工资信息，包括工资条生成、工资发放、薪资调整等功能。
3. 员工状态管理模块：用于管理员工的在职、离职、请假、加班等状态，以及相关的审批流程和记录。
4. 查询统计管理模块：提供员工信息、工资信息、员工状态等方面的查询和统计功能，以便管理者进行数据分析和决策。

4.3.2 软件配置项之间的静态关系：

图示如下：

员工信息管理模块 <--- 查询统计管理模块

工资管理模块 <--- 查询统计管理模块

员工状态管理模块 <--- 查询统计管理模块

4.3.3 软件配置项分配的 CSCI 需求和 CSCI 级设计决策：

1. 员工信息管理模块 CSCI 需求：

- 实现员工信息的增删改查功能，包括基本信息、联系方式等。
- 提供员工信息的导入导出功能，支持 Excel、CSV 等格式。

2. 员工信息管理模块 CSCI 决策：

- 使用 Django 框架实现员工信息管理模块，利用 Django 的 ORM 实现数据库操作。
- 基于 Bootstrap 等前端框架设计用户界面，以提升用户体验。

3. 工资管理模块 CSCI 需求：

- 实现工资条的生成和发放功能，支持自定义工资条模板和薪资发放周期。
- 提供薪资调整功能，支持按月份或按百分比调整薪资。

4. 工资管理模块 CSCI 决策：

- 使用 Django 框架实现工资管理模块，利用 Django 的表单功能实现工资条的生成和调整。
- 使用 Celery 实现工资发放的异步任务，以提升系统性能和用户体验。

5. 员工状态管理模块 CSCI 需求：

- 实现员工在职、离职、请假、加班等状态的管理功能，支持审批流程和记录。
- 提供相关状态变更的通知功能，以便及时通知相关人员。

6. 员工状态管理模块 CSCI 决策：

- 使用 Django 框架实现员工状态管理模块，利用 Django 的模型和表单功能实现状态变更的记录和审批。
- 使用 Django 的信号框架实现状态变更的通知功能，以便及时通知相关人员。

4.3.4 软件配置项的开发状态/类型：

1. 员工信息管理模块和工资管理模块：这两个模块都是基于 Django 框架实现的，属于 Web 应用的管理模块。由于 Django 框架成熟稳定，相关技术栈和应用案例丰富，因此这两个模块的开发状态是相对成熟的。

2. 员工状态管理模块：由于涉及审批流程和记录，需要较为复杂的业务逻辑和数据库设计，因此开发状态相对较新，需要进行更多的研究和实践。

3. 查询统计管理模块：查询统计管理模块属于数据分析和报表展示类模块，开发状态相对成熟，可以基于现有的数据分析库和报表工具进行快速实现。

4.3.5 CSCI 计划使用的计算机硬件资源：

处理器能力：需要一定的处理器能力以支持数据的快速处理和计算。

内存容量：需要足够的内存以存储大量的员工信息、工资信息和状态记录。

输入/输出设备能力：需要支持用户通过 Web 界面进行数据的输入和输出。

辅存容量：需要足够的辅存容量以支持大文件的上传和处理。

通信/网络设备能力：需要支持用户在上传和下载文件时进行网络通信，以及在系统内部进行数据交互和传输。

4.3.6 软件配置项之间的动态关系：

软件配置项之间的动态关系是通过相互调用和交互来实现的。下面是每个模块与其他模块之间存在的一些动态关系：

1. 员工信息管理模块：
 - 可以调用查询统计管理模块以生成员工信息的相关报表和统计数据。
 - 可以与工资管理模块和员工状态管理模块交互，以更新员工的工资信息和在职状态。
2. 工资管理模块：
 - 可以调用查询统计管理模块以生成工资信息的相关报表和统计数据。
 - 可以与员工信息管理模块和员工状态管理模块交互，以获取员工的基本信息和在职状态。
3. 员工状态管理模块：
 - 可以调用查询统计管理模块以生成员工状态的相关报表和统计数据。
 - 可以与员工信息管理模块和工资管理模块交互，以更新员工的基本信息和工资信息。
4. 查询统计管理模块：
 - 可以调用员工信息管理模块、工资管理模块和员工状态管理模块以获取相关数据。
 - 可以与用户交互界面模块交互，以展示生成的报表和统计数据。

4.3.7 软件配置项的质量要求：

每个软件配置项在设计和实现时都需要满足一定的质量要求，以确保整个软件系统的稳定性和可靠性。下面是对每个模块的质量要求的描述：

1. 员工信息管理模块：
 - 确保员工信息的准确性和完整性，避免数据冗余和错误。
 - 提供高效的数据导入导出功能，支持多种格式和大数据量。
 - 提供可靠的数据备份和恢复功能，以保证数据安全和可靠性。
2. 工资管理模块：
 - 确保工资条的准确性和及时性，避免薪资计算错误和漏发情况。
 - 提供高效的工资发放和调整功能，支持大规模员工的薪资管理。
 - 提供可靠的审批流程和记录，以保证数据的完整性和安全性。
3. 员工状态管理模块：
 - 确保状态变更的准确性和及时性，避免因状态错误导致的业务问题。
 - 提供可靠的审批流程和记录，以保证状态变更的合法性和安全性。
 - 提供实时的状态变更通知和提醒功能，以便相关人员及时处理。

4. 查询统计管理模块：

提供高效的数据查询和统计功能，支持大规模数据的处理和分析。

提供可视化的报表展示和定制功能，以满足不同用户的需求。

提供可靠的数据导出和分享功能，支持多种格式和大数据量。

总之，每个模块的质量要求都是相对高的，我们在设计和实现时需要进行充分的测试和优化，以确保其在系统中的可靠性和稳定性。

4.4 执行概念

本条应描述软件配置项间的执行概念。为表示软件配置项之间的动态关系，即 CSCI 运行期间它们如何交互的，本条应包含图示和说明，(若适用)包括执行控制流、数据流、动态控制序列、状态转换图、时序图、配置项之间的优先关系、中断处理、时间/序列关系、异常处理、并发执行、动态分配与去分配、对象/进程/任务的动态创建与删除和其他的动态行为。

4.4 执行概念

在软件系统中，各个软件配置项之间存在着动态的交互和执行关系，以实现系统的功能和性能要求。下面将分别描述各个软件配置项之间的执行概念。

4.4.1 员工信息管理模块

执行控制流：

当用户需要查询或修改员工信息时，员工信息管理模块接收用户请求，并根据请求类型执行相应的操作。例如，当用户需要查询员工信息时，控制流会执行查询操作，并返回相应的结果。

数据流：

用户通过 Web 界面输入员工信息，数据流经过前端验证后传输至员工信息管理模块进行处理。在处理过程中，员工信息管理模块将数据存储至数据库中，并返回处理结果给用户。

动态控制序列：

用户输入员工信息请求 → 数据传输至员工信息管理模块 → 数据验证和处理 → 结果返回给用户

4.4.2 工资管理模块

执行控制流：

当用户需要生成工资条或调整工资信息时，工资管理模块接收用户请求，并根据请求类型执行相应的操作。例如，当用户需要生成工资条时，控制流会执行工资条生成操作，并将结果返回给用户。

数据流：

用户通过 Web 界面输入工资信息，数据流经过前端验证后传输至工资管理模块进行处理。在处理过程中，工资管理模块将数据存储至数据库中，并返回处理结果给用户。

动态控制序列：

用户输入工资信息请求 → 数据传输至工资管理模块 → 数据验证和处理 → 结果返回给用户

4.4.3 员工状态管理模块

执行控制流：

当用户需要变更员工状态或查询员工状态时，员工状态管理模块接收用户请求，并根据请求类型执行相应的操作。例如，当用户需要变更员工状态时，控制流会执行状态变更操作，并将结果返回给用户。

数据流：

用户通过 Web 界面输入状态变更请求，数据流经过前端验证后传输至员工状态管理模块进行处理。在处理过程中，员工状态管理模块将数据存储至数据库中，并返回处理结果给用户。

动态控制序列：

用户输入状态变更请求 → 数据传输至员工状态管理模块 → 数据验证和处理 → 结果返回给用户

4.4.4 查询统计管理模块

执行控制流：

当用户需要查询统计信息或生成报表时，查询统计管理模块接收用户请求，并根据请求类型执行相应的操作。例如，当用户需要生成员工信息报表时，控制流会执行报表生成操作，并将结果返回给用户。

数据流：

用户通过 Web 界面输入查询统计请求，数据流经过前端验证后传输至查询统计管理模块进行处理。在处理过程中，查询统计管理模块从数据库中获取数据，并根据用户需求生成报表或统计结果，然后将结果返回给用户。

动态控制序列：

用户输入查询统计请求 → 数据传输至查询统计管理模块 → 数据查询和统计处理 → 结果生成并返回给用户

4.4.5 软件配置项之间的动态关系示意图：

用户请求（输入）--> 员工信息管理模块 --> 数据库（员工信息）
用户请求（输入）--> 工资管理模块 --> 数据库（工资信息）
用户请求（输入）--> 员工状态管理模块 --> 数据库（员工状态信息）
用户请求（输入）--> 查询统计管理模块 --> 数据库（各类信息）

6 其他动态行为:

- **并发** 各个之间可以同时处理多个用户请求，以提高系统的并发性能。
- - **中断** 在系统运行过程中，可能会出现各种异常情况，需要进行相应的中断处理以保证系统的稳定性和可靠性。
- - **异常** 当系统出现异常情况时，需要进行相应的异常处理，例如数据验证失败、数据库连接中断等。
- - **动态分配与释放**：系统运行过程中会动态分配和释放资源，以满足用户请求的处理和系统运行的需要。
- - **对任务的动态创建与删除**：系统对象、进程或任务会根据用户请求的动态创建和删除，以进行相应的业务处理和系统管理。

4.5 接口设计

在软件系统中，接口设计是确保不同软件配置项之间能够有效通信和交互的关键部分。本节将描述软件配置项的接口特性，包括与其他软件配置项以及外部实体（如系统、配置项和用户）之间的接口。

4.5.1 接口标识与接口图

每个接口都被赋予了一个唯一的项目标识符，以便识别和区分不同的接口。接口标识符包括名称、编号、版本和文档引用等信息，用于标识接口实体，例如软件配置项、系统、配置项和用户。以下是接口图，描述了各个软件配置项之间的接口关系：

[员工信息管理模块] <--> [工资管理模块]
[员工信息管理模块] <--> [员工状态管理模块]
[员工信息管理模块] <--> [查询统计管理模块]

4.5.x (接口的项目唯一标识符)

4.5.2 员工信息管理模块接口设计

- a. 优先级： 高
- b. 接口类型： 实时数据传输
- c. 数据元素特性：

- 1. 名称/标识符： 项目唯一标识符： IF001

- a. 非技术名称： 员工信息

- 2. 数据类型： 结构化数据
- 3. 大小与格式： 可变长度的字符串
- 4. 计量单位： 无
- 5. 范围或可能值的枚举： N/A
- 6. 准确度与精度： 高
- 7. 优先级、时序、频率、容量、序列和其他约束： 高优先级，实时更新
- 8. 保密性与私密性约束： 中等
- 9. 来源与接收者： 来源于用户，接收者为员工信息管理模块

- d. 数据集合体特性：

- 1. 名称/标识符： 项目唯一标识符： IF002

- a. 非技术名称： 员工信息集合

- 2. 数据元素及结构： 包括员工姓名、工号、部门、职位等
- 3. 媒体及媒体上的结构： N/A
- 4. 显示和其他输出的视听特性： N/A
- 5. 数据集合体之间的关系： N/A
- 6. 优先级、时序、频率、容量、序列和其他约束： 高优先级，实时更新
- 7. 保密性与私密性约束： 中等
- 8. 来源与接收者： 来源于用户，接收者为员工信息管理模块

- e. 通信方法特性：

- 1. 项目唯一标识符： IF003
- 2. 通信链路/带宽/频率/媒体： HTTP 协议，互联网连接

- 3. 消息格式化: JSON 格式
 - 4. 流控制: 无
 - 5. 数据传输率、周期或非周期和传送间隔: 实时传输
 - 6. 路由、寻址及命名约定: 通过 URL 寻址
 - 7. 传输服务: 高优先级传输
 - 8. 安全性/保密性/私密性考虑: 使用 HTTPS 进行加密传输
- f. 协议特性:
- 1. 项目唯一标识符: IF004
 - 2. 协议的优先级/层: 高
 - 3. 分组: N/A
 - 4. 合法性检查、错误控制、恢复过程: 通过 HTTP 状态码进行错误控制和恢复
 - 5. 同步: 无
 - 6. 状态、标识和其他报告特性: 使用 HTTP 头部信息传递状态信息
- g. 其他特性:
- 物理兼容性: 支持各种常见的网络连接方式, 如有线、无线等。
-

5 CSCI 详细设计

5.1 员工信息管理模块

5.1.1 概述

该模块负责管理员工的基本信息、学历信息、婚姻状况以及其他相关信息。它包括信息录入、修改、查询和统计等功能, 输出包括完整的员工信息记录和统计分析报告。

5.1.2 软件配置项设计中的约束、限制或非常规特征

对于员工信息的录入, 需要进行字段验证和数据一致性检查, 以确保数据的准确性和完整性。对于大规模员工信息的管理, 需要进行数据库优化和索引设计, 以提高查询和统计的效率。

员工信息管理模块需要遵循数据保护法规和公司隐私政策，对敏感信息进行特殊处理和保护。

5.1.3 编程语言和使用理由

Python 是该模块所使用的编程语言，有丰富的库和框架，强大的数据处理能力及灵活可扩展，庞大的开发者社区支持。这些优势使 Python 成为开发信息管理系统的理想选择，提高了开发效率和系统的性能。适合开发复杂的人事管理系统。

5.1.4 过程式命令列表

add_user: 该命令用于录入新员工的个人信息，包括基本信息、学历信息等。该命令调用信息录入界面和数据库写入模块来完成该任务。

reset_user: 该命令用于修改现有员工的个人信息。该命令调用信息编辑界面和数据库更新模块来完成该任务。

find_user: 该命令用于查看特定员工的个人信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

change_usr: 该命令用于修改特定员工的个人信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

delete_usr: 该命令用于删除特定员工的个人信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

find_all_usr: 该命令用于寻找所有员工的信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

find_usr_by_id: 该命令用于按编号寻找员工信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

find_usr_by_marriage: 该命令用于按婚姻情况寻找员工信息。该命令调用信息查看界面和数据库查询模块来完成该任务。

5.1.5 输入输出和数据元素说明

输入数据元素

文本内容：包括中文和英文文本。表示员工信息；

输出数据元素

检索信息：包括中文和英文文本，表格，图片等；

5.1.6 逻辑说明

该软件配置项执行启动时，其内部起作用的条件：

- 系统接收到新增、修改或查询员工信息的请求。
- 系统已经通过了用户身份验证和访问权限检查。

把控制交给其他软件配置项的条件：

- 当员工信息管理模块完成信息录入或修改后，将数据传递给数据库存储模块进行持久化保存。
- 当需要查看或统计员工信息时，将查询请求传递给数据库查询模块。

对每个输入的响应及响应时间，包括数据转换、重命名和数据传送操作：

- 输入为员工信息的字段数据，需要进行数据验证和格式转换以确保数据质量。
- 响应时间取决于数据库操作的速度和网络延迟，通常在数秒内完成。

该软件配置项运行期间的操作序列和动态控制序列，包括：

a) 序列控制方法：

- 根据用户请求类型（新增、修改、查询等）执行相应的操作流程。
- 对输入数据进行验证，确保数据的完整性和准确性。
- 执行数据库操作，如插入、更新、删除或查询记录。
- 返回操作结果给用户界面。

b) 该方法的逻辑与输入条件，如计时偏差、优先级赋值：

- 数据验证和格式转换是实时进行的，以确保数据的一致性。
- 数据库操作根据操作类型和系统负载动态调整优先级，以优化性能。

c) 数据在内存中的进出：

- 输入的员工信息数据首先存储在内存中的数据结构中，如对象或数据模型。
- 处理后的数据通过数据库接口进出内存，与数据库进行交互。

d) 离散输入信号的感知，以及在软件配置项内中断操作之间的时序关系：

- 软件配置项内部监听用户操作和系统事件，如用户提交表单或定时任务触发。
- 中断操作可能包括用户取消操作或系统错误，需要适当处理这些情况以保持系统稳定。

5.2 员工工资管理模块

5.2.1 概述

该模块负责管理员工的工资信息，包括工资项目配置、工资计算、工资条生成和工资发放等功能。它确保员工的工资计算准确无误，并能够生成详细的工资条供员工查询。

5.2.2 软件配置项设计中的约束、限制或非常规特征

对于工资项目的配置，需要进行灵活的设计，以适应不同员工的薪资结构和变化。

对于工资计算，需要考虑各种计算规则，如加班费、扣除项、税率等。

工资管理模块需要遵循当地的法律法规，确保工资计算的合法性和合规性。

5.2.3 编程语言和使用理由

搜索模块采用 Python 语言进行开发，原因如下：

- Python 具有良好的可读性和可维护性，代码简洁，易于理解和修改。
- Python 拥有丰富的第三方库和框架，可快速构建高效的搜索引擎。
- Python 支持多线程和协程，可以提高搜索效率和并发处理能力。
- Python 适用于各种应用场景，包括 Web 开发、数据处理和科学计算等。

5.2.4 过程式命令列表

config_salary: 该命令用于配置工资项目，包括基本工资、奖金、扣款等。该命令调用工资项目配置界面和数据库写入模块来完成该任务。

Calculate_Salary: 该命令用于计算员工的工资。该命令调用工资计算模块和数据库查询模块来完成该任务。

Generate_salary: 该命令用于生成员工的工资条。该命令调用工资条生成模块和数据库查询模块来完成该任务。

send_salary: 该命令用于发放员工的工资。该命令调用工资发放模块和数据库更新模块来完成该任务。

find_salary: 该命令用于查看特定员工的工资条。该命令调用工资条查看界面和数据库查询模块来完成该任务。

5.2.5 逻辑说明

该软件配置项执行启动时，其内部起作用的条件：

- 系统接收到工资项目配置、工资计算或工资发放的请求。
- 系统已经通过了用户身份验证和访问权限检查。

把控制交给其他软件配置项的条件：

- 当工资计算完成后，将计算结果传递给工资条生成模块。
- 当工资发放完成后，将发放记录传递给工资记录模块。

对每个输入的响应及响应时间，包括数据转换、重命名和数据传送操作：

- 输入为员工工资相关的数据，需要进行数据验证和格式转换以确保数据质量。
- 响应时间取决于数据库操作的速度和网络延迟，通常在数秒内完成。

该软件配置项运行期间的操作序列和动态控制序列，包括：

a) 序列控制方法：

- 根据用户请求类型（工资项目配置、工资计算、工资发放等）执行相应的操作流程。
- 对输入数据进行验证，确保数据的完整性和准确性。
- 执行数据库操作，如插入、更新、删除或查询记录。
- 返回操作结果给用户界面。

b) 该方法的逻辑与输入条件，如计时偏差、优先级赋值：

- 数据验证和格式转换是实时进行的，以确保数据的一致性。

- 数据库操作根据操作类型和系统负载动态调整优先级，以优化性能。
- c) 数据在内存中的进出：
 - 输入的员工工资数据首先存储在内存中的数据结构中，如对象或数据模型。
 - 处理后的数据通过数据库接口进出内存，与数据库进行交互。
- d) 离散输入信号的感知，以及在软件配置项内中断操作之间的时序关系：
 - 软件配置项内部监听用户操作和系统事件，如用户提交表单或定时任务触发。
 - 中断操作可能包括用户取消操作或系统错误，需要适当处理这些情况以保持系统稳定。

5.3 查询与统计

5.3.1 概述

该模块提供对员工信息的查询和统计分析功能，以便管理人员能够快速获取所需信息并了解员工情况。它包括多条件查询、信息汇总和报表生成等功能。

5.3.2 软件配置项设计中的约束、限制或非常规特征

对于查询功能，需要进行高效的数据库设计和查询优化，以提高查询速度和响应时间。
对于统计功能，需要实现灵活的统计方法和报表生成，以适应不同管理需求。
查询与统计模块需要考虑数据安全和隐私保护，确保只有授权用户能够访问敏感信息。

5.3.3 编程语言和使用理由

Python 是该模块所使用的编程语言，主要是因为其强大的数据处理和数据分析库（如 Pandas、NumPy）以及丰富的可视化工具（如 Matplotlib、Seaborn），适合开发高效的查询与统计功能。

5.3.4 过程式命令列表

PerformQuery: 该命令用于根据用户输入的条件执行查询操作。该命令调用查询界面和数据库查询模块来完成该任务。

GenerateReport: 该命令用于生成统计报表。该命令调用统计模块和报表生成模块来完成该任务。

ExportData: 该命令用于将查询结果或统计报表导出为特定格式的文件。该命令调用数据导出模块来完成该任务。

5.3.5 逻辑说明

该软件配置项执行启动时，其内部起作用的条件：

- 系统接收到查询或统计的请求。

- 系统已经通过了用户身份验证和访问权限检查。

把控制交给其他软件配置项的条件：

- 当查询或统计操作完成后，将结果传递给用户界面或报表生成模块。

对每个输入的响应及响应时间，包括数据转换、重命名和数据传送操作：

- 输入为查询条件或统计参数，需要进行数据验证和格式转换以确保数据质量。
- 响应时间取决于数据库操作的速度和网络延迟，通常在数秒内完成。

该软件配置项运行期间的操作序列和动态控制序列，包括：

a) 序列控制方法：

- 根据用户请求类型（查询、统计等）执行相应的操作流程。
- 对输入数据进行验证，确保数据的完整性和准确性。
- 执行数据库操作，如查询、汇总或生成报表。
- 返回操作结果给用户界面。

b) 该方法的逻辑与输入条件，如计时偏差、优先级赋值：

- 数据验证和格式转换是实时进行的，以确保数据的一致性。
- 数据库操作根据操作类型和系统负载动态调整优先级，以优化性能。

c) 数据在内存中的进出：

- 输入的查询条件或统计参数首先存储在内存中的数据结构中，如对象或数据模型。
- 处理后的数据通过数据库接口进出内存，与数据库进行交互。

d) 离散输入信号的感知，以及在软件配置项内中断操作之间的时序关系：

- 软件配置项内部监听用户操作和系统事件，如用户提交查询请求或定时任务触发。
- 中断操作可能包括用户取消操作或系统错误，需要适当处理这些情况以保持系统稳定。

6 需求的可追踪性

6.1 CSCI 需求可追踪性

6.1.1 查询模块

- CSCI 需求：用户查询员工基本信息，工资条，绩效，违规记录；系统能够正确及时给出反馈。
- 可追踪性：该模块可以追踪到 CSCI 需求中的“员工信息查询，工资绩效查功能”。

6.1.2 员工信息修改模块

- CSCI 需求：员工信息修改功能功能。要求普通员工能够修改个人基本信息。

- 可追踪性：该模块可以追踪到 CSCI 需求中的“员工信息修改功能”。

6.1.3 员工管理模块

- CSCI 需求：员工状态管理功能。要求管理员用户能够管理用户状态，登记员工工资，绩效，违规信息。
- 可追踪性：该模块可以追踪到 CSCI 需求中的“员工状态管理功能”。

6.1.4 提交申请模块

- CSCI 需求：提交申请功能。要求员工可以提交一些申请，以及附件。
- 可追踪性：该模块可以追踪到 CSCI 需求中的“提交申请功能”。

6.2 CSCI 需求到它被分配给软件配置项的可追踪性

6.2.1 查询功能

- 软件配置项：查询模块。
- 可追踪性：该 CSCI 需求可以追踪到软件配置项中的“查询模块”。

6.2.2 员工信息修改功能

- 软件配置项：员工信息修改模块。
- 可追踪性：该 CSCI 需求可以追踪到软件配置项中的“员工信息修改模块”。

6.2.3 员工管理功能

- 软件配置项：员工管理模块。
- 可追踪性：该 CSCI 需求可以追踪到软件配置项中的“员工管理模块”。

6.2.4 提交申请功能

- 软件配置项：提交申请模块。
- 可追踪性：该 CSCI 需求可以追踪到软件配置项中的“提交申请模块”。

7 注解

7.1 背景信息

本文档描述的是一个基于 web 的人事管理系统的软件结构设计。该系统的主要功能是人事管理，员工信息查询修改，工资绩效查询，员工状态管理，提交申请等。

7.2 术语和定义

以下是本文档中使用的一些术语和定义：

- CSCI：计算机软件组件。
- SDK：软件开发工具包，是一组用于开发软件的工具集合。
- API：应用程序接口，是一组定义了软件组件之间交互方式的规范。

7.3 缩略语列表

以下是本文档中使用的一些缩略语及其定义：

- SAD：软件结构设计说明文档。
- FAR：可行性分析（研究）报告文档。
- SRS：软件需求规格文档。
- PMS：人事管理系统。
- UI：用户界面，是指用户与应用程序进行交互的界面。
- API：应用程序接口，是一组定义了软件组件之间交互方式的规范。

附录

附录可用来提供那些为便于文档维护而单独出版的信息(例如图表、分类数据)。为便于处理，附录可单独装订成册。附录应按字母顺序(A, B 等)编排。